IN THE UNITED STATES DISTRICT COURT
FOR THE MIDDLE DISTRICT OF TENNESSEE
NASHVILLE DIVISION

CONCORD MUSIC GROUP, INC., et al., Plaintiffs, v. Anthropic PBC, Defendant.	Case No. 3:23-cv-01092 Chief Judge Waverly D. Crenshaw, Jr. Magistrate Judge Alistair Newbern

DECLARATION OF BEN Y. ZHAO

I, **Ben Y. Zhao**, hereby declare pursuant to 28 U.S.C. § 1746:

1. I am the Neubauer Professor of Computer Science at the University of Chicago, and a Fellow of the Association for Computing Machinery ("ACM"). I submit this declaration in connection with Plaintiffs' Motion for a Preliminary Injunction filed by Plaintiffs Concord Music Group, Inc., Capitol CMG, Inc., Universal Music Corp., Songs of Universal, Inc., Universal Music – MGB NA LLC, Polygram Publishing, Inc., Universal Music – Z Tunes LLC, and ABKCO Music, Inc. (collectively, "Publishers"), in their lawsuit against Defendant Anthropic PBC.

2. My statements set forth below are based on my specialized knowledge, education, and experience, as applied to the facts and circumstances of this case. If called upon, I would and could competently testify as to the matters contained herein.

I. My Assignment

3. Publishers have asked me to carefully consider Anthropic's current artifact AI system known as "Claude," and to make determinations of its functionality, its likely training data and training process, and in the context of potential guardrails to limit or eliminate the harms of reproducing (in whole or in part) Plaintiffs' lyrics, what mechanisms are possible, currently

deployed, and their fundamental limitations. More specifically, I am asked to answer these following questions:

- Has Anthropic copied Publishers' song lyrics and used those copies to train Claude?
- How critical is this copying and training process to the general functionality of Claude?
- Can Anthropic implement guardrails that prevent Claude from returning Publishers' lyrics in its queries, and what are the limitations/failures of these mechanisms?
- Is Anthropic capable of training a large language model that is guaranteed to return none of Publishers' compositions?

4. In my work on this matter, I have reviewed and utilized a variety of materials, including a) publicly accessible versions of Claude, b) recorded transcripts of prior interactions with Claude referenced in the Declaration of BC Guardian, and c) current research literature on generative AI models, particularly on large language models. I am also relying on my background in artificial intelligence, deep neural networks (DNNs), and adversarial machine learning generally. I note that Claude is a proprietary system developed by Anthropic, which has not released to the public the underlying source code, model weights, or comparable material to permit a detailed inspection of its operation. Researchers have given Anthropic / Claude 2 a relatively low transparency score of 36%.[1]

5. In discussing my analysis in this declaration, I do so in a manner geared toward a general audience. If I am asked to write an Expert Report at a later stage of this case, I can elaborate on the points made herein in more technical terms with detailed citations, but I assume a higher level of discussion is more helpful to the Court at this stage.

[1] *The Foundation Model Transparency Index*, CTR. FOR RSCH. ON FOUND. MODELS, https://crfm.stanford.edu/fmti/ (last visited Nov. 13, 2023).

II. My Background

6. I am the Neubauer Professor of Computer Science at the University of Chicago, a Fellow of the Association for Computing Machinery, and co-director of the UChicago Security, Algorithms, Networks, and Data (SAND) Lab. I teach and mentor undergraduate and PhD students and conduct research. Over the years, I have led numerous projects with $30M funded by the National Science Foundation, Department of Defense, Department of State, and industry research groups. Prior to starting my faculty career 20 years ago, I received my PhD from UC Berkeley, and my Bachelor of Science from Yale University.

7. My research lies at the intersection of machine learning and security, with additional topics covering data-driven systems and human computer interaction. My work is published regularly at the most selective publication venues in computer security and machine learning and has received numerous best paper awards, distinguished paper awards, most-influential paper awards and the Internet Defense Prize. My early work created the research area now known as distributed hash tables, and those systems are now used in most large cloud computing systems today. More recently, my projects established the first defenses against backdoor attacks on deep neural networks and developed adversarial techniques to protect creative artists against misuses of generative AI models. My 190 publications have garnered over 35,000 citations. My full curriculum vitae is attached to this declaration as **Exhibit A**.

III. Context

8. Since its inception, Anthropic has branded itself as an AI company prioritizing AI safety and minimizing harm. In their core views, they list "critically evaluating the potential societal impacts of our work" as a "key pillar of our research."[2] Their approach to AI safety, billed

[2] *See Research Principles*, ANTHROPIC, https://www.anthropic.com/research (last visited Nov. 11, 2023).

"Constitutional AI," is designed to give an AI system a set of principles (i.e., a "constitution") against which it can evaluate its own outputs. Anthropic offers its main product Claude, a large language model ("LLM"), both as a standalone chatbot interface, as well as an API for integration into other AI products and applications.

IV. Summary of Conclusions

9. Based on transcripts of interactions with Claude described in the Complaint and evidence listed in the Declaration of Dan Seymour, it is clear that Anthropic has trained Claude on song lyrics that include Publishers' music. It is also clear from Claude's outputs that there have been some guardrails put in place to limit its output of potentially infringing content, but those guardrails easily fail for reasons that are both unique to Claude and indicative of all similar models. In the short run, improving guardrails might stem some inclusion of Publishers' music in Claude's outputs but ultimately this is an incomplete and ineffective fix since guardrails are easily circumvented and inherently limited. In the longer term, Anthropic will inevitably release a new version of Claude, which will be re-trained on a new dataset. When that occurs, Anthropic should exclude Publishers' compositions from its training material. As discussed below, this is a reasonable solution to the problem presented by Anthropic's choices.

V. What is AI?

A. Traditional vs. Generative AI

10. Artificial Intelligence (AI) systems are built to flexibly perform tasks that involve a wide variety of novel situations and that ordinarily would be expected to require human intelligence to accomplish. "Traditional" AI, is designed to perform a specific task in a predictable way to user-supplied inputs based on pre-determined rules. These systems mainly fall under the broader category of discriminatory classification systems because their primary goal is to

recognize complex patterns inherent in multi-faceted data and to produce decisions as a result. For example, a drug-discovery AI system might take as input a new hypothetical chemical compound structure, and based on its training, produce a classification result that predicts whether the compound will have a specific prescriptive impact on a particular disease or pathology. In other examples, a trained deep neural network (DNN) classifier may take in photographic images of skin lesions and predict which samples are likely cancerous, and a different DNN might take audio samples of a human voice and produce a classification of the emotional state of the speaker and if they are telling the truth. Many well-known AI applications are built using traditional AI such as spam filters in email, generating recommendations on e-commerce platforms, self-driving cars, data mining, virtual assistants like Siri, and chess-playing programs. These architectures and applications existed years before the current arrival of generative AI applications.

11. Generative AI strives to generate content by combining existing materials together using patterns it derives from training materials. It utilizes complex machine learning architectures to extract correlation patterns between text descriptions and specific content samples (text, images, audio) that match those text descriptions. Sufficient training on large datasets produces the generative model, the encapsulation of a massive compilation of correlations between text prompt and content, largely stored as conditional probabilities. These probabilities can then guide the reproduction of content that have the highest chances of matching any text prompt given by the user of the generative model.

12. Today's leading generative AI systems can be used to generate text, images, audio visual works, music and more. Regardless of their specific task or architecture, generative AI systems are trained on very large datasets of content according to the form of content they are intended to generate and input they are designed to accept. The volume and variety of these training

data is critical, because the model can only produce high quality content on a specific topic once it sees sufficient training samples relevant to that specific topic.

13. LLMs are currently among the most popular and impactful applications of generative AI today. An LLM is an algorithm that uses massively large datasets to predict and generate content in response to a prompt. They can be deployed in different settings to perform a variety of natural language processing ("NLP") tasks, including text summarization, language translation, and writing long form text documents such as short essays, poems, song lyrics and software code. Text-generative LLMs are trained to respond to queries in a small number of languages, so users can input prompts and obtain results in natural language without the need to write computer code.

14. All AI models are based on a particular architecture (the design and structure of the model). LLMs typically utilize a "transformer" architecture, a type of machine learning algorithm.[3] Among other features, a transformer is comprised of an encoder, which reads and processes the input data, and a decoder that generates predictions. Transformers work on predicting high probability text "tokens" given some contextual information, where the context is often measured in number of tokens of memory. The more powerful and complex the model, the more tokens it can capture in its contextual memory to guide its prediction of meaningful responses to incoming user questions and requests.

B. LLM Training

15. The creation of an LLM is a multi-stage project. Preliminary steps include defining the object of the model (will it be a chatbot, a translation tool, etc.), designing the architecture, and related tasks. This initial work is followed by data-intensive steps of training the defined model.

[3] Ashish Vaswani et al., *Attention Is All You Need* (Aug. 2, 2023), arXiv:1706.03762v7, https://arxiv.org/pdf/1706.03762.pdf.

In preparation for training, the creator of an LLM gathers text material, typically in massive quantity and from a variety of sources and copies it to a central repository. Frequently, the text is obtained from websites, and the LLM creator may either copy it directly from them, use archives of such text compiled by third parties, or use a mix of sources.

16. The next step is "cleaning" the raw aggregated text according to various criteria, such as removing certain sorts of recognized duplicates (for example, copies of the same webpage). In addition, content will be removed that is inconsistent with the LLM creator's objectives for the model; examples could include offensive material such as pornography, software source code, and certain languages or character sets. The result of this process is a cleaned version of the dataset that then can be used for training; this cleaned version represents an additional copy.

17. After the data is cleaned, the text is converted into "tokens." On average, a token is three quarters of a word, meaning a token is a combination of a few characters (including punctuation) with a ratio of about 4 tokens per 3 words.

18. After the dataset is cleaned and tokenized, the LLM is trained. The first step is called pre-training, during which the model is trained to predict the next token/word in a string. For example, it will be given the sentence, "The dog wagged its ___" and learn to predict the next word as "tail". Or "I like ice ___" where the answer could be "cones", "cream" or various other choices. Through pre-training, the model will learn which of these is the best choice from context. Pre-training also involves "weighting" the dataset. In this process, the trainer directs the model to use some parts of the dataset more often than others due to differences in perceived quality of the material, with portions engaged multiple times and others not at all. Ultimately the model will learn the most common patterns in the grammar, syntax, and semantics of a given language. This process requires no supervision (i.e. labeling or metadata describing the contents of the text).

Rather, the model recognizes patterns in the training data to infer these content labels automatically.

19. After a model is pre-trained, the next step is "fine-tuning," during which the LLM is specialized to perform a particular task through supervised learning.[4] This process typically utilizes "reinforcement learning," where computers and humans rate the output supplied by the model in response to a given prompt, adjusting the values of the model's parameters to steer it to prefer certain sorts of responses and avoid others. This enables the model to produce more accurate and tailored responses. The most popular version of this training is commonly referred to as RLHF, or "reinforcement learning with human feedback."

C. LLM Outputs

20. The LLM enhanced with reinforcement learning may be offered as a service available for human interaction on a website as a "chatbot" or available for use by other software through an "application programming interface" ("API"). In either case, the user supplies an input known as a "prompt," the model processes the prompt in a step known as "inference", and then the service returns a response or "completion," also called "outputs."

21. In generating outputs, LLMs are entirely dependent on their training datasets. A recent paper from Google DeepMind confirmed what most in the community already knew, that transformers cannot generalize beyond their pretraining data.[5] Put another way, the entirety of the functionality of these LLMs is defined by their training data, and they are unable to generalize beyond it.

[4] *Glossary,* ANTHROPIC, https://docs.anthropic.com/claude/docs/glossary (last visited Nov. 13, 2023).

[5] Steve Yadlowsky et al., *Pretraining Data Mixtures Enable Marrow Model Selection Capabilities in Transformer Models* (Nov. 1, 2023), arXiv: 2311.00871, https://arxiv.org/pdf/2311.00871.pdf.

22. In addition, once an LLM is trained on a given piece of content, it can use that content for any purpose. Other than in the weighting of data sources during training, LLMs do not differentiate among the sources of training data in creating outputs. It may use the tokens representing song lyrics to generate lyrics, poetry, nonfiction essays, marketing materials, movie scripts, or anything else. Because there are few category labels on incoming training data, there are few if any mechanisms in place to prevent content of one type being used to produce responses to unrelated prompts on a different topic.

23. Beyond those steps taken during the cleaning and training process, in an effort to constrain model responses, LLM creators may implement "guardrails" to limit undesired output. These guardrails can be applied as filters on prompts to prevent problematic generation in the first place. They also can be implemented as filters on completions to catch undesirable output before it is delivered to users. In either case, the guardrails exist outside the underlying model as opposed to being inherent within it. Guardrails can be enforced reactively to limit model output of certain text memorized from training data, but research demonstrates that there are consistent and reliable ways to bypass existing guardrails, and they are not a panacea to limit all potential abuses.[6]

24. While LLMs are not, in general, a simple database of the training content, it is possible for certain content to be "memorized" by LLMs. Studies have shown memorization by LLMs is more likely for content that is repeated frequently in the training dataset and that larger LLMs memorize more than smaller ones.[7] As with most machine learning models, the large space of potential prompts or queries can only be trained with massive amounts of training data covering

[6] Andy Zou et al., *Universal and Transferable Adversarial Attacks on Aligned Language Models* (July 27, 2023), arXiv: 2307.15043, https://arxiv.org/pdf/2307.15043.pdf. Proceedings of ICML, 2023.

[7] Kent K. Chang et al., *Speak, Memory: An Archaeology of Books Known to ChatGPT/GPT-4* (Oct. 20, 2023), arXiv: 2305.00118v2, https://arxiv.org/pdf/2305.00118.pdf; Antonia Karamolegkou et al., *Copyright Violations and Large Language Models* (Oct. 20, 2023), arXiv: 2310.13771, https://arxiv.org/pdf/2310.13771.pdf.

the broad span of prompts. While model training becomes more robust with increased volume of training on a particular topic, the opposite also holds. Words, proper names, or relatively unusual sentences or phrases that occur rarely in training data will lead LLMs to "overfit" on them, making the model more likely to repeat them (and phrases or text blocks that contain them) verbatim given the correct leading sequence or prompts. For training data that is extremely popular (and therefore is represented in training data over and over as different copies), this overfitting effect is magnified and reinforced. This leads to the observed effect that rare token sequences are more easily extracted from LLM models.[8] In fact, current research from Google establishes the understanding that LLMs memorize text, and memorization grows with 1) increases in capacity of a model, 2) the number of times an example has been duplicated, and 3) the number of tokens of context used to prompt the model.[9]

D. Updating LLMs

25. An LLM can only utilize the information in the dataset on which it was trained, which makes that information fixed in time as of the moment the dataset was collected. These models therefore get "stale" and there is enormous pressure on commercial LLMs to regularly update their model to incorporate the most recent content available into the dataset and make any needed technical adjustments.

26. When a developer decides to release a new version of an LLM, it cannot simply add the latest information to the existing dataset. Instead, the dataset is collected anew, and then trained from scratch in the process described above.

[8] Nicholas Carlini et al., *Extracting Training Data from Large Language Models* (June 15, 2021), arXiv: 2012.07805v2, https://arxiv.org/pdf/2012.07805.pdf.

[9] Nicholas Carlini et al., *Quantifying Memorization Across Neural Language Models* (Mar. 6, 2023), arXiv: 2202.07646, https://arxiv.org/pdf/2202.07646.pdf.

27. Accordingly, if a developer wants to exclude certain content from its dataset, it cannot cause the model to "forget" material used in its training—training data cannot be removed once tokenized and used to set model weights. But the LLM can be relaunched without that particular content as part of its dataset when the next version of the LLM is released.

VI. LLMs and Copyright

28. Respecting copyright while building an LLM model is not impossible. It can be accomplished through a combination of implementing effective guardrails and avoiding training AI models on unlicensed materials. Model developers can avoid scraping from sources known to contain unlicensed copyrighted material such as Books3, which I understand is a massive database of nearly 200,000 books downloaded from pirate sources. Model developers can also look out for copyright identifiers such as the copyright symbol ©, authors' or publishers' name, copyright dates, and other related indicia of copyright protection. Some of that material may be licensed for public use through Creative Commons licenses, but that should be clearly indicated. Developers can also utilize databases of copyrighted works as a "blacklist" to exclude from the training data any instance of those works found in that database. Finally, developers can seek licenses from content owners to use their works as training material.

29. Developers can also attempt to limit the display of copyrighted material in outputs by implementing content-based blockers (rules that restrict the model from including specific text either in prompts or outputs) as guardrails. Unfortunately, guardrails are imperfect.

30. First, content-based blockers are known to be fragile, and can be bypassed by a user with sufficient knowledge of LLMs. Recent research published this year at the top machine learning conference (ICML) demonstrates fundamental flaws in the way transformer language models are built, such that there are deterministic ways to compute non-sensical characters, which

when added to a prompt, will bypass prompt blockers, and allow the query to be answer truthfully by the LLM.[10] This work also demonstrates that these vulnerabilities are fundamental to the transformer architecture and are applicable to all commercial LLM models. It is also not a property or weakness that can be easily addressed. Transformer architecture is inherently vulnerable to these attacks and simply cannot be "debugged" to prevent infringing outputs in all cases. Accordingly, while guardrails have their place in a well-designed LLM, they are inherently limited.

31. Third, output-based guardrails do not address the use of unlicensed works as part of the dataset in the first place.

VII. Claude

A. Overview

32. Claude is a system with an LLM at its core and an interactive interface ("chatbot") built on top. The current version is Claude 2 (released July 2023). Claude is available as a web version for noncommercial users and as an API for commercial users. The API version is available with Claude 2 and Anthropic's latest offering, Claude Instant.

33. Claude's primary difference from other LLMs is its reliance on what it calls "Constitutional AI," a list of rules or principles that the model slowly approximates using a combination of supervised learning and reinforcement learning that Anthropic calls "RL from AI

[10] Andy Zou et al., *Universal and Transferable Adversarial Attacks on Aligned Language Models* (July 27, 2023), arXiv: 2307.15043, https://arxiv.org/pdf/2307.15043.pdf; Zico Kolter, Keynote Presentation at the 2nd ICML Workshop on New Frontiers in Adversarial Machine Learning: Adversarial Attacks on Aligned LLMs (July 28, 2023).

Feedback" or (RLAIF).[11] In comparison, ChatGPT relies more heavily on Reinforcement Learning with Human Feedback (RLHF).[12]

34. The Claude web version is available as a paid subscription-based version and a more limited free version. The free version of Claude can be accessed by anyone who creates an (anonymous) account that is verified only with a text message sent to a mobile phone number.

35. Claude 2 and earlier Claude models are available to commercial customers to be incorporated into their own software, products, and systems through an API. Claude models are currently already integrated into a variety of products, including online question & answer platforms (Quora), group messaging applications (Slack), Internet search engines (DuckDuckGo), and note-taking applications (Notion). To interact with the Claude chatbot, users enter a prompt into the customer software. Claude then processes the users' prompt and generates a response which is then delivered back to the user through the customer software. This process is invisible to the ultimate end user.

B. Training/Copying Copyrighted Material

36. Anthropic has the ability to decide what content it will use to train Claude and whether it will disclose that information publicly. Prior to March 2023, LLM developers were more transparent about the contents of their training datasets.[13] Anthropic has chosen not to make its training dataset information public. Anthropic has stated only that "Claude models are trained on a proprietary mix of publicly available information from the Internet, datasets that we license from third party businesses, and data that our users affirmatively share or that crowd workers

[11] Yuntao Bai et al., *Constitutional AI: Harmlessness from AI Feedback* (Dec. 15, 2022), arXiv: 2212.08703, https://arxiv.org/pdf/2212.08073.pdf.

[12] Long Ouyang et al., *Training Language Models to Follow Instructions with Human Feedback* (Mar. 4, 2022), arXiv: 2203.02155, https://arxiv.org/pdf/2203.02155.pdf; *GPT-4 Technical Report*, OPENAI (Mar. 15, 2023), arXiv: 2303.08774, https://arxiv.org/pdf/2303.08774.pdf.

[13] Kyle Barr, *GPT-4 is a Giant Black Box and Its Training Data Remains a Mystery*, GIZMODO (Mar. 17 2023), https://gizmodo.com/chatbot-gpt4-open-ai-ai-bing-microsoft-1850229989.

provide," and that the text on which Claude 2 was trained continues through early 2023 and is 90 percent English language.[14]

37. Anthropic's limited disclosures make clear that in research it conducted in 2021 (laying the groundwork for Claude), it has relied heavily on datasets (e.g., the "Common Crawl" dataset) that include large amounts of copyrighted and unlicensed material, likely including from popular lyrics websites, to train its AI models.[15]

38. It is also clear based on the outputs BC Guardian has collected from Claude that Anthropic has trained Claude on song lyrics that include Publishers' works. *See* Seymour Decl. Exs. B, C. We would obviously not see perfect copies of song lyrics in response to a query unless the system copied and stored those lyrics. The probability of a randomized LLM generating specific lyrics comprised of hundreds of words in a single song, let alone hundreds of songs, without having trained on them, is astronomically low.

39. In addition, while LLMs are designed to mimic fluent speech and text composition, the perfect or near-perfect reproduction of song lyrics cannot have been generated naturally by the LLMs themselves. Song lyrics often use non-standard words or verbiage that are found nowhere else, making them extremely unlikely to be predicted by an LLM. For example, Will Smith's song "Fresh Prince of Bel-Air" includes not only proper nouns like Bel-air, but also non-standard contractions like 'maxin' and 'makin', words rarely seen outside of this particular song's lyrics. These unique words, when combined with perfect verbatim phrases that match the entire song, leave little doubt that the model is reproducing the lyrics after being trained on them. In other

[14] *Model Card and Evaluations for Claude Models* 2, ANTHROPIC, https://www-files.anthropic.com/production/images/Model-Card-Claude-2.pdf.
[15] *A General Language Assistant as a Laboratory for Alignment* 8, 27, ANTHROPIC (Dec. 9, 2021), https://arxiv.org/pdf/2112.00861.pdf.

words, the reproduction would be extremely unlikely if the model had not been exposed to copies of the lyrics during training. *See* Complaint ¶ 74.

40. In addition, song lyrics will naturally be a small portion of the overall dataset in an LLM model given the percentage of lyrics relative to other written work on the internet. However, because the structure and content of song lyrics are uncommon and specific to that genre of writing, and given Claude's readiness to display lyrics as outputs, I would expect song lyrics were weighted relatively heavily in Claude's training.

41. Further, as discussed above, once an LLM is trained on a given piece of content, it can then use that content for any purpose. Publishers' lyrics can be used not just to deliver full versions of those lyrics but to generate works that include parts of the original work blended with different text. Those AI-generated works can be suggested as distinct musical compositions, poems, short stories, marketing pitches, or any other form of written work—but particularly those requiring the unique structure of song lyrics. Anthropic's unauthorized use of Publishers' lyrics also enable Claude's generation of outputs generally by adding to its overall lexicon as part of the larger training dataset. In other words, song lyrics have value to the Claude system beyond simply being available to display verbatim to users who request them explicitly.

C. Infringing Outputs

42. My interactions with Claude revealed that completions from Claude are variable and uncontrolled. Some unpredictability is expected given the role of randomized token selection in transformer architectures. However, even for transformer-architecture models, Claude's seeming willingness to generate infringing content and its ability to avoid doing so seems highly uncontrolled and random. As shown in the results from BC Guardian, Claude reproduced verbatim or near-verbatim versions of the lyrics to 500 songs. Seymour Decl. Ex. B. But getting consistent results of this sort from Claude is unlikely. My own research with Claude demonstrated that Claude

would quickly respond to a request for the lyrics to "Roar" by Katy Perry, by producing verbatim lyrics, but refuse a request for lyrics to Will Smith's "Fresh Prince of Bel-Air", citing copyright issues. Sometimes rewording a denied request would produce verbatim song lyrics. At other times, simply repeating the request will produce the desired result. The identical request will be denied one day, then allowed to proceed on a different day. These results are attached as **Exhibit B**.

43. Claude also routinely produces outputs that contain Publishers' lyrics, even when not explicitly asked for those lyrics. For example, when asked to write a poem in the style of Louis Armstrong or to create something on the death of Buddy Holly, Claude reproduced large portions of the original lyrics relevant to those prompts. Complaint ¶¶ 73, 79.

44. It is apparent Claude's guardrails are ineffective and porous. This is likely due in part to inherent weaknesses in guardrails, as discussed above. In this regard, a recent study demonstrated Claude 2's vulnerability.[16] The authors explain that Claude 2 has some keyword blockers in place to prevent it from outputting offensive and dangerous content. Nevertheless, using simple contextual redefinitions or keyword macros allowed the researchers to bypass these protections from Claude, and in one example, successfully get Claude to produce a detailed plan on how to destroy humanity!

45. Guardrails also have limited impact when the prompt is not explicitly seeking infringing content. Recall that Claude declines to produce the lyrics to Will Smith's "Fresh Prince of Bel-Air." However, this "guardrail" has no effect if there is no direct reference to the title of the song. Given a request to "Write me a poem about moving from Philadelphia to Bel-Air," Claude responds by producing the entire lyrics to "Fresh Prince of Bel-Air," presented as a poem titled "Fresh Out of Philly." See **Exhibit B**.

[16] Andy Zou et al., *Universal and Transferable Adversarial Attacks on Aligned Language Models* (July 27, 2023), arXiv: 2307.15043 https://arxiv.org/pdf/2307.15043.pdf.

46. Nevertheless, even for LLM models generally, Claude's guardrails seem to be particularly ineffective and could be strengthened further.

VIII. Remedy

A. Short Term

47. In the short term, Anthropic can improve the guardrails on infringing outputs from Claude. Although guardrails are ultimately imperfect, they can certainly function better than they currently do on Claude. Anthropic should be able to write prompt-blocking code to prevent Claude from returning Publishers' song lyrics in response to explicit or implicit requests for them.

B. Long Term

48. As discussed above, post-training guardrails cannot be the long-term solution to the use of Publishers' compositions in training.

49. First, the copying of Publishers' work as training data is itself an unauthorized reproduction that Anthropic can only address by relaunching Claude without including Publishers' works. Second, the only surefire method to eliminate all model use of such text as an output is to remove it proactively from the training dataset.

50. As discussed above, Publisher's compositions cannot be removed from the current version of Claude. But that data can be removed from Claude's dataset the next time Anthropic relaunches the model. This is perfectly reasonable. Commercial LLM models such as Claude are periodically re-trained and re-launched from scratch. For example, in updating Claude to Claude 2, Anthropic had to create and train an entirely new dataset.

51. When it undertakes its next update, Anthropic can train Claude on a dataset that does not include Publishers' compositions. This is the only way to ensure that Claude will not be able to include this content in its outputs.

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct to the best of my personal knowledge and belief.

Executed in Chicago, IL, this 16th day of November, 2023.

Ben Y. Zhao

CERTIFICATE OF SERVICE

I hereby certify that on November 16, 2023, I authorized the electronic filing of the foregoing with the Clerk of the Court using the CM/ECF system, which will send notification of such filing to the following:

Aubrey B. Harwell III
Nathan C. Sanders
Olivia R. Arboneaux
NEAL & HARWELL, PLC
1201 Demonbreun Street, Suite 1000
Nashville, TN 37203
tharwell@nealharwell.com
nsanders@nealharwell.com
oarboneaux@nealharwell.com

Andrew Gass
Joe Wetzel
LATHAM & WATKINS, LLP
505 Montgomery St., Suite 2000
San Francisco, CA 94111
andrew.gass@lw.com
joe.wetzel@lw.com

Allie Stillman
LATHAM & WATKINS, LLP
1271 Avenue of the Americas
New York, NY 10020
alli.stillman@lw.com

Sy Damle
LATHAM & WATKINS, LLP
555 Eleventh Street, NW, Suite 1000
Washington D.C. 20004
sy.damle@lw.com

s/ Steven A. Riley